Amendments to the Specification

Please replace the paragraph beginning on page 5, line 27 and ending on page 6, line 6 with the following amended paragraph:

Turning now to FIG. 2, a conceptual illustration of the remote computer system 100 associated with the present invention is shown in accordance with an embodiment of invention. The central intelligence of the remote computer system 100 is a baseboard 102, or "motherboard", which is a printed circuit board to which a multitude of components or devices may be connected by way of a system bus 112 or other electrical communication path (not shown). In an embodiment, these components include, without limitation, a controller such as the baseboard management controller (hereinafter, "BMC") 104, a central processing unit (hereinafter, "CPU") 108, a network adapter 122, a system memory and an Input/Output module 110. The system memory may include ~~including~~ a random access memory ("RAM") 106 and a read-only memory ("ROM") 107.

Please replace the paragraph beginning on page 7, line 25 and ending on page 8, line 12 with the following amended paragraph:

The firmware of the BMC 104 adheres to the Intelligent Platform Management Interface (IPMI) industry standard for system monitoring and event recovery. The IPMI standard is well-known to those of ordinary skill in the industry, and therefore not described in detail herein. Rather, revision 1.1 of the IPMI Specification, version 1.5, release date February 20, 2002, is incorporated by reference. The BMC 104 monitors operating and performance-related parameters received from various components of the computer system 100 in order to determine whether an "event" is occurring within the system 100. In addition, it is also contemplated that the BMC 104 may also be connected to and receive data from components in communication with the network ~~interface~~ adapter 122 or from components connected to a bus (e.g., system bus 112) other than the management bus 130. For example, the BMC 104 may receive data from the local computer 201 through the network adapter 122 over the network 18. The BMC 104 may also be utilized by the remote computer 100 to send data to the local computer 201 via the LAN adapter 122 over the network 18. The BMC 104 may also include a shared memory which may be accessed by various components in the computer system 100. In one embodiment, the shared memory may include a send buffer 132 and a receive buffer 135 for sending and receiving data between various components in the remote computer system 100 or alternatively in the local computer system 201 via the network adapter 122 over the network 18.

Please replace the paragraph on page 8, beginning on line 13 and ending on line 24 with the following amended paragraph:

The Input/Output module 110 is used as a communication medium between any number and type of peripheral devices and the system bus 112. Communications destined for the CPU 108, the BMC 104 or any other component coupled to the system bus 112 and issued by a peripheral device must therefore pass through the Input/Output module 110 to the system bus 112 and then to the necessary component. As shown in FIG. 2, the Input/Output module 110 is connected a mass storage device 14 for storing an operating system 16 and application programs ~~30~~31. The operating system 16 comprises a set of programs that control operations of the remote computing system 100 and allocation of resources. The set of programs, inclusive of certain utility programs, also provide a graphical user interface to the user. An application program is software that runs on top of the operating system software and uses computer resources made available through the operating system to perform application specific tasks desired by the user.

Please replace the paragraph on page 9, beginning on line 9 and ending on line 22 with the following amended paragraph:

The Input/Output module 110 is also connected to a keyboard controller 115 which controls the transfer of data between the computer system and a user input module 116, e.g., a keyboard. The Input/Output module 110 is also connected to a video display adapter 113 which enables the display of video data (i.e., text and/or graphics) on a display unit 114. It should be understood that the video display adapter 113 may offer both text and graphics video modes for display on the display unit 114. In text mode only text (i.e., ASCII) characters may be displayed, while in graphics mode the display unit may display any bit-mapped image. In one embodiment, the video display adapter 113 is set to text mode for displaying text or ASCII characters on the display unit ~~113~~114. The video display adapter ~~114~~ 113 also contains a video text data memory or video buffer 140 for temporarily storing one or more lines of video data while displayed on the display unit 114. It will be appreciated that the local computer system 201 described briefly above with respect to FIG. 1 may include many or all of the elements described above relative to the remote computer system 100.

Please replace the paragraph on page 10, beginning on line 18 and ending on line 31 with the following amended paragraph:

As is further known to those skilled in the art, a system timer interrupt is one of several hardware interrupts in a computer system which ~~signal~~ signals the CPU that an event has occurred. In a typical ~~computer~~ IBM-compatible computer system, the system timer interrupt (INT-08) is generated 18.2 times per second, or every 54.9 milliseconds. In one embodiment, the program code in the redirection program module 160 may be copied into a memory area in RAM (i.e., the interrupt vector table) occupied by the program code for calling the default interrupt handler routine for the system timer interrupt. Thus, when the system timer interrupt is generated by the remote computer system 100, the program code in the redirection program module 160 is executed before the default interrupt handler routine. After the instructions contained within the program code in the redirection program module 160 have been executed, the redirection program module 160 will then call the default timer interrupt handler routine. A block diagram showing the above-described procedure is shown in FIG. 4, and will be described in greater detail in the description of FIG. 4, below.

Please replace the paragraph on page 11, beginning on line 1 and ending on line 15 with the following amended paragraph:

The logical operations 300 continue from operation 305 to operation 310 where the redirection program module 160 retrieves the memory address location and the size of the send buffer 132 in the BMC 104 which will be utilized to store redirected video text data prior to being sent from the BMC 104 to the local computer 201. The logical operations 300 then continue from operation 310 to operation 315 where the redirection program module 160 accesses the video buffer 140 in the video display adapter 113 and selects a line of stored video text data. For example, the redirection program module 160 may select the first line of video text data from among twenty-five lines of data in the video buffer 140. As discussed in the description of FIG. 2 above, the video buffer 140 is utilized by the video display adapter 113 for temporarily storing one or more lines of video text data prior to displaying the video text data on the display unit 114. In one embodiment, the number of lines capable of being stored in the video buffer 140 may be equal to the number of lines capable of being displayed on the display unit 114. A block diagram showing illustrative contents of the video buffer 140 is shown in FIG. 5, and will be described in greater detail in the description of FIG. 5, below.

Please replace the paragraph on page 11, beginning on line 16 and ending on line 25 with the following amended paragraph:

The logical operations 300 continue from operation 315 to operation 320 where the redirection program module 160 calculates a checksum for the selected line of video text data in the video buffer 140. A checksum is defined as a unique numerical value which is computed based on the characters in the selected line of video text data. In one embodiment, the checksum may be calculated from the identity of the characters in the line of video text data (e.g., "A," "B," etc.), the position of the characters, and a text attribute associated with the characters (e.g., bold, italics, underline, etc.). Once a checksum for a line of video data is calculated, it is saved in a shared memory location in the RAM 106. It will be appreciated that the number of saved checksums may be equal to the number of lines of video text data stored in the video buffer 140.

Please replace the paragraph on page 12, beginning on line 7 and ending on line 17 with the following amended paragraph:

If at operation 340 it is determined that there are additional lines of video text data stored in the video buffer 140, the logical operations 340 continue to operation 345 where the redirection program module 60 selects the next line of video text data. The logical operations 300 then return to operation 320. If however, at operation 340 it is determined that there are no more lines of video text data in the video buffer 140, then the logical operations 300 continue from operation 340 to operation 350 where the redirection program module ~~350~~ 160 instructs the BMC 104 to send the contents of the send buffer 132 (i.e., the changed lines of video text data) through the network adapter 122 to the local computer 201 via the network 18. The logical operations 300 then continue from operation 350 to operation 355 where the instructions contained in the redirection program module 160 call the default system timer interrupt handler.

Please replace the paragraph beginning on page 11, line 26 and ending on page 12, line 6 with the following amended paragraph:

The logical operations 300 then continue from operation 320 to operation 325 where the redirection program module 160 compares the just calculated checksum for the selected line of video text data to any previously saved checksums in the RAM 106. If the just calculated checksum is not equal to any of the previously saved checksums, then the selected line of video text data is changed video text data and the logical operations 300 continue from operation 325 to operation 330 where the redirection program module 160 communicates with the BMC 104 to determine if the send buffer 132 is full. If at operation 330, the redirection program module 160 determines that the send buffer 132 is not full, then the redirection program module 160 stores the changed line of video text data in the send buffer 132 at operation 335. The logical operations 300 then continue from operation ~~330~~ 335 to operation 340 where the redirection module 160 determines whether there are any more lines of video text data stored in the video buffer 140.